

Sampling Strategies for Fast Updating of Gaussian Markov Random Fields

D. Andrew Brown*

Christopher S. McMahan[†]

February 21, 2017

Abstract

Gaussian Markov random fields are popular for modeling temporal or spatial dependence in large areal datasets due to their ease of interpretation, simple specifications, and computational convenience afforded by conditional independence and the consequent sparse precision matrices needed for random variable generation. Using such models inside a Markov chain Monte Carlo algorithm requires repeatedly simulating random fields. This is a nontrivial issue, especially when the full conditional precision matrix depends on parameters that change at each iteration. Typically in Bayesian computation, Gaussian Markov random fields are updated jointly in a block Gibbs sampler or one location at a time in a single-site sampler via the available full conditional distributions. The former approach leads to quicker convergence by updating correlated variables all at once, while the latter avoids solving large matrices. Efficient algorithms for sampling Markov random fields have become the focus of much recent research in the machine learning literature, much of which can be useful to statisticians. We briefly review recently proposed approaches with an eye toward implementation for statisticians without expertise in numerical analysis or advanced computing. In particular, we consider a version of block sampling in which the underlying graph can be cut so that conditionally independent sites are all updated together. This algorithm allows a practitioner to parallelize the updating of a subset locations or to take advantage of ‘vectorized’ calculations in a high-level language such as R. Through both simulation and real data application, we demonstrate computational savings that can be achieved versus both traditional single-site updating and block updating, regardless of whether the data are on a regular or irregular lattice. We argue that this easily-implemented sampling routine provides a good compromise between statistical and computational efficiency when working with large datasets.

Key Words: Bayesian computation, Cholesky factorization, chromatic Gibbs sampling, conditional autoregressive model, graph coloring, Markov chain Monte Carlo

*Corresponding author, Department of Mathematical Sciences, Clemson University, Clemson, SC, USA 29634-0975, Email: ab7@clemson.edu

[†]Department of Mathematical Sciences, Clemson University, Clemson, SC, USA

1 Introduction

Markov random fields are now a cornerstone upon which many statistical models are built for the analysis of large-scale correlated data. Awareness of these models was raised after the seminal work of Besag (1974). Since the turn of the century, they have become popular for modeling temporally- or spatially-dependent areal data due to their interpretability and computational tractability afforded by the conditional independence induced by the Markov property. This property is a particularly important concern for modern Markov chain Monte Carlo (MCMC; Gelfand and Smith, 1990) methods. Indeed, the ease with which Markov random fields can be incorporated into a Gibbs sampling algorithm (Geman and Geman, 1984) has no doubt contributed to their popularity in Bayesian statistics. Markov random fields are now viewed as critical tools in a variety of challenging applications, including disease mapping (Waller et al., 1997), image deblurring (Fox and Norton, 2016), positron emission tomography (Higdon, 1998), functional magnetic resonance imaging (Brown et al., 2014), and gene microarray (Xiao et al., 2009; Brown et al., 2017).

Perhaps the most important type of Markov random field is the Gaussian case. Gaussian Markov random fields (GMRFs; Rue and Held, 2005), as the name implies, are simply Markov random fields in which the conditional distribution of each (scalar) random variable is Gaussian. In the statistics literature they are more commonly referred to as conditionally autoregressive (CAR; Banerjee et al., 2015) models. GMRFs are unique in that they are specified by explicitly defining the precision (inverse covariance) matrix instead of building a covariance matrix via a covariance function as in Gaussian process modeling (Schabenberger and Gotway, 2005). This approach may result in an improper distribution such as the intrinsic autoregressive model (IAR; Besag and Kooperberg, 1995). Such improper distributions thus can only be used as priors in Bayesian models. In addition, GMRFs do not usually yield stationary processes due to a so-called “edge effect” in which the marginal variances of the random variable vary by location. Corrections can be made to yield a stationary process such as a periodic boundary assumption (Fox and Norton, 2016) or algorithmic specification of the precision matrix (Dempster, 1972). Sometimes the effect can simply be ignored with little effect on inference (Besag and Kooperberg, 1995). Efforts have been made to use GMRFs to approximate Gaussian processes with specified covariance functions (e.g., Rue and Tjelmeland, 2002; Song et al., 2008; Lindgren et al., 2011), but much work remains to be done in this direction.

Belonging to the Gaussian class of distributions, GMRFs are the most widely studied Markov random fields. (See Rue and Held (2005) for an overview of relevant work.) This of course includes techniques for efficiently sampling from GMRFs. While both single-site and block samplers are straightforward in principle, in practice they can be problematic when working with extremely high-dimensional datasets. For instance, block sampling involves Cholesky factorizations of large precision matrices. While the neighborhood structure of a GMRF prior induces sparsity which can sometimes be exploited to economize such calculations, conditional posterior precision matrices arising in Bayesian models may depend on parameters that change in each iteration of an MCMC algorithm, making repeated calculations extremely time consuming. On the other hand, single-site samplers work by only considering scalar random variable updates. In addition to being inherently more loop-intensive than block samplers, single-site samplers are known to exhibit slow convergence when the variables are highly correlated since they move very slowly through the support of the target distribution (Carlin and Louis, 2009).

There is an apparent trade-off between computational and statistical efficiency when using high-dimensional GMRFs inside an MCMC algorithm. These competing goals have led to recent innovations in alternative sampling approaches for GMRFs. Much of this work has been done in the machine learning literature, but it is of interest to practicing statisticians, as well. While some of these approaches require considerable expertise in numerical analysis or message passing interface

(MPI) protocol, we have found that others are relatively easy to implement and hence can be quite useful for statisticians.

In this paper, we discuss some recently proposed approaches for efficient sampling of high-dimensional Markov random fields with particular attention to the Gaussian case. We discuss also the two dominant approaches among statisticians, namely block updating and single-site updating. Rather than focusing on theoretical convergence rates or an otherwise overall “best” approach, we view these approaches through the lens of a practitioner looking for easily implemented yet reasonably efficient algorithms that strike a good balance between computational and sampling efficiency. Specifically, we have found that the recently proposed “chromatic Gibbs sampler” (Gonzalez et al., 2011) is easy to implement and is competitive with or even able to dramatically improve upon well-known block updating strategies. This sampling algorithm is in a sense a hybrid approach between single-site and block updating. It allows a practitioner to parallelize sampling the random field and to take advantage of ‘vectorized’ calculations in a high-level language such as R (R Core Team, 2016) without requiring extensive expertise in numerical analysis. To the best of our knowledge, this work is the first time chromatic Gibbs sampling has been directly compared to Cholesky-based block sampling of Gaussian random fields.

The remainder of this paper is organized as follows: In Section 2, we review GMRFs along with standard sampling approaches as well as some more novel approaches appearing in the machine learning literature. We focus in particular on chromatic sampling and compare it to block updating and single-site sampling. In Section 3, we numerically compare the performance of single-site sampling, block updating, and the chromatic approach in a simulation study using a simple Bayesian model with spatial random effects on a regular lattice. We consider also a real data example motivated by climatology in which the areal data are arranged on an irregular lattice, demonstrating remarkable improvements there, as well. We conclude in Section 4 with a discussion of our findings and practical suggestions for those wishing to implement MCMC for large-scale data with underlying GMRF structure.

2 MCMC Sampling for Gaussian Markov Random Fields

Consider a Gaussian Markov random field (GMRF) $\mathbf{x} = (x_1, \dots, x_n)^T$, where x_i is the realization of the field at node i , $i = 1, \dots, n$. The density of \mathbf{x} is given by

$$\begin{aligned} \pi(\mathbf{x} \mid \boldsymbol{\mu}) &\propto \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{Q}(\mathbf{x} - \boldsymbol{\mu})\right) \\ &\propto \exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{b}^T \mathbf{x}\right), \end{aligned} \tag{1}$$

where $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\mathbf{b} = \mathbf{Q}\boldsymbol{\mu}$. If \mathbf{Q} is of full rank and hence nonsingular, then this distribution is proper and the normalizing constant is $(2\pi)^{-n/2} \det(\mathbf{Q})^{1/2}$. Intrinsic GMRFs are such that the \mathbf{x} variables have linear constraints so that \mathbf{Q} is rank deficient. In this case, we may define the density with proportionality constant $(2\pi)^{-(n-k)/2} \det^*(\mathbf{Q})^{1/2}$, where $n - k$ is the rank of \mathbf{Q} and $\det^*(\cdot)$ is a generalized determinant in that it is the product of the $n - k$ non-zero eigenvalues of \mathbf{Q} (Hodges et al., 2003; Rue and Held, 2005). Such improper GMRF models are common in Bayesian disease mapping as they are easily interpretable and usually result in proper posterior distributions. They arise also in linear inverse problems in which \mathbf{Q} is obtained as the discrete Laplacian over an image (Bardsley, 2012).

GMRFs may be specified according to an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} indicates nodes (i.e., the vertices) and $\mathcal{E} = \{(i, j) : i \sim j\}$ is the edge set where $i \sim j$ if and only if node i is

connected to node j . The precision matrix \mathbf{Q} is determined by $(\mathbf{Q})_{ij} \neq 0$ if and only if $(i, j) \in \mathcal{E}$. Specifying the density through the precision matrix \mathbf{Q} instead of an explicit covariance matrix induces a Markov property in the random field (Rue and Held, 2005, Theorem 2.2). For any node i , $x_i \mid \mathbf{x}_{(-i)} \stackrel{d}{=} x_i \mid \mathbf{x}_{\mathcal{N}(i)}$, where $\mathbf{x}_{(-i)} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)^T$, $\mathcal{N}(i) = \{j : (i, j) \in \mathcal{E}\}$ is the neighborhood of node i , and $\mathbf{x}_{\mathbf{A}} := (x_i, i \in \mathbf{A})^T$ for some index set \mathbf{A} . In other words, x_i is conditionally independent of the rest of the field given its neighbors. Most GMRFs assume that each node has relatively few neighbors, resulting in a sparse precision matrix \mathbf{Q} . The sparsity of the precision matrix combined with the intuitive and easily-interpretable Markov property has led to GMRFs being widely used to model dependence in large-scale areal data.

With the need to model extremely large datasets with nontrivial correlation has come the need for efficient sampling techniques whereby posterior distributions of fully Bayesian models can be simulated. This has led to recent work in the machine learning and numerical analysis literature on efficient simulation from high-dimensional Markov random fields, both Gaussian and otherwise. When periodic boundary conditions on $\mathbf{x} \in \mathbb{R}^n$ can be assumed (i.e., each x_i has the same number of neighbors, including the edge nodes), Fox and Norton (2016) note that the sampling problem can be diagonalized via the Fast Fourier Transform (with complexity $\mathcal{O}(n \log n)$), whence a sample can be drawn by solving a system in $\mathcal{O}(n)$ operations. They propose reducing the total number of draws from the conditional distribution of \mathbf{x} by using a “marginal-then-conditional” sampler in which the MCMC algorithm operates by completely collapsing over \mathbf{x} and subsequently sampling \mathbf{x} using only the approximately independent draws of the hyperparameters obtained from a full MCMC run on their marginal distribution. In many applications, though, the periodic boundary assumption may not be realistic, and sampling from the marginal distribution of hyperparameters could itself be challenging. To avoid the computational difficulties associated with full GMRFs, Cai et al. (2013) propose using a pairwise graphical model as an approximate GMRF for high-dimensional data imputation without specifying the precision matrix directly. The authors admit, however, that this procedure is very hard to implement (Cai, 2014, p. 7). In cases where we are given \mathbf{Q} and \mathbf{b} in (1) with the goal of estimating $\boldsymbol{\mu}$, Johnson et al. (2013) express the Gibbs sampler as a Gauss-Seidel iterative solution to $\mathbf{Q}\boldsymbol{\mu} = \mathbf{b}$, facilitating the “Hogwild” parallel algorithm of Niu et al. (2011) in which multiple nodes are updated simultaneously without locking the remaining nodes. In the Gaussian case, Johnson et al. (2013) prove convergence to the correct solution when the precision matrix \mathbf{Q} is symmetric diagonally dominant. Motivated by Johnson et al. (2013), Cheng et al. (2015) use results from spectral graph theory to propose a parallel algorithm for approximating a set of sparse factors of \mathbf{Q}^l , $-1 \leq l \leq 1$, in nearly linear time. They show that it can be used to construct *iid* samples from an approximate distribution. This is opposed to a Gibbs sampler, which produces approximately independent samples from the correct distribution. Similar to the Gauss-Seidel splitting considered by Johnson et al. (2013), Liu et al. (2015) propose an iterative approach to approximating a draw from a GMRF in which the corresponding graph is separated into a spanning tree and the missing edges, whence the spanning tree is randomly perturbed and used as the basis for an iterative linear solve.

The aforementioned algorithms can be difficult to implement and require substantial knowledge of graph theory, numerical analysis, and MPI programming. This makes such approaches inaccessible to many practicing statisticians who nevertheless need to work with large random fields. In addition, they are iterative routines for producing even a single draw from an approximation to the target distribution. This feature makes them less appealing for users who work in R. It is well-known that loops should only be used with care in R, to avoid repeated data type interpretation and memory overhead. In light of these difficulties while still faced with the problem of efficient updating of GMRFs inside a larger MCMC algorithm, we consider in the remainder of this Section

the two classic approaches for sampling from a GMRF, as well as a more recently proposed parallel sampler which we believe is easy to implement and amenable for vectorization in R.

2.1 Block and Single-Site Gibbs Sampling

In this Section, it is helpful to distinguish between sampling \mathbf{x} directly from a *prior* GMRF and from the full conditional distribution of \mathbf{x} derived from an hierarchical Bayesian model with a GMRF prior on \mathbf{x} . For an unconditional GMRF, the distribution is of the form

$$\mathbf{x} \sim N(\boldsymbol{\mu}, \mathbf{Q}^{-1}), \quad (2)$$

where $\boldsymbol{\mu}$ and \mathbf{Q} are generally unrelated. (Often, we take $\boldsymbol{\mu} = \mathbf{0}$.) On the other hand, when drawing from the full conditional distribution inside a Gibbs sampler, the sampling target is of the form $\mathbf{x} \sim N(\mathbf{Q}_p^{-1}\mathbf{b}, \mathbf{Q}_p^{-1})$, where $\mathbf{Q}_p \neq \mathbf{Q}$ is an updated precision matrix. For example, in a typical linear model $\mathbf{y} \sim N(\mathbf{A}\mathbf{x}, \boldsymbol{\Sigma})$ with \mathbf{A} fixed and $\mathbf{x} \sim N(\boldsymbol{\mu}, \mathbf{Q}^{-1})$, standard multivariate normal theory yields

$$\begin{aligned} \mathbf{x} \mid \mathbf{y}, \boldsymbol{\Sigma} &\sim N(\mathbf{Q}_p^{-1}\mathbf{b}, \mathbf{Q}_p^{-1}), \\ \mathbf{Q}_p &= \mathbf{A}^T \boldsymbol{\Sigma}^{-1} \mathbf{A} + \mathbf{Q} \\ \mathbf{b} &= \mathbf{A}^T \boldsymbol{\Sigma}^{-1} \mathbf{y} + \mathbf{Q}\boldsymbol{\mu}. \end{aligned} \quad (3)$$

Two competing approaches to updating GMRFs inside a Markov chain Monte Carlo algorithm are so-called single site sampling in which individual sites are updated one at a time using the available full conditional distributions, and block Gibbs sampling in which the entire random field is updated all at once via sampling from a known multivariate Gaussian distribution induced by the GMRF. Block sampling is known to improve the convergence of Gibbs samplers in the presence of correlated variables by allowing the chain to move more quickly through the sample space (Liu et al., 1994; Rue and Held, 2005). The drawback is in the manipulation and solution of large covariance matrices necessary for both random variable generation and evaluation of the likelihood in a Metropolis-Hastings algorithm. Alternatively, single site updating uses the conditional distributions of each scalar random variable on the lattice, thus avoiding large matrix computations. In single-site sampling, though, sampling efficiency is sacrificed as updating a group of correlated parameters one at a time results in very slow exploration of the parameter space, slowing convergence of the Markov chain.

An appealing feature of GMRFs is the ability to specify the distribution of \mathbf{x} through a complete set of full conditional distributions, $\{p(x_i \mid \mathbf{x}_{(-i)}) : i = 1, \dots, n\}$. For instance, we can assume each $x_i \mid \mathbf{x}_{(-i)} \sim N(\eta_i, \sigma_i^2)$, with $\eta_i = \mu_i + \sum_{j \sim i} c_{ij}(x_j - \mu_j)$ and $\sigma_i^2 > 0$, where c_{ij} are specified weights such that $c_{ij} \neq 0$ if and only if $i \sim j$ and $c_{ii} = 0$ for all i . Specification of a Markov random field through these so-called local characteristics was pioneered by Besag (1974), after which such models came to be known as *conditional autoregressive (CAR) models*. Besag (1974) used Brook's Lemma and the Hammersley-Clifford Theorem to establish that the set of full conditionals collectively determine a joint density, provided a positivity condition holds among \mathbf{x} . In this case, we have that

$$(\mathbf{Q})_{ij} = \begin{cases} -c_{ij}/\sigma_i^2, & i \neq j \\ 1/\sigma_i^2, & i = j \end{cases},$$

where the condition $\sigma_j^2 c_{ij} = \sigma_i^2 c_{ji}$, for all i, j , is necessary to ensure symmetry of \mathbf{Q} . The ease with which these full conditional distributions can be incorporated into a Gibbs sampling algorithm has led to a dramatic increase in the popularity of CAR models over the past fifteen years or so (Banerjee et al., 2015).

Input: Mean μ , precision matrix Q .
Output: Draw x from a $N(\mu, Q^{-1})$ distribution.

- 1 Find the Cholesky factor $Q = LL^T$
- 2 Sample $z \sim N(0, I)$
- 3 Solve $L^T v = z$
- 4 Compute $x = \mu + v$
- 5 **Return** x

Algorithm 1: Sampling from a typical prior GMRF model (Rue and Held, 2005).

Input: Mean factor b , precision matrix Q_p .
Output: Draw x from a $N(Q_p^{-1}b, Q_p^{-1})$ distribution.

- 1 Find the Cholesky factor $Q_p = LL^T$
- 2 Solve $Lw = b$
- 3 Solve $L^T \mu = w$
- 4 Sample $z \sim N(0, I)$
- 5 Solve $L^T v = z$
- 6 Compute $x = \mu + v$
- 7 **Return** x

Algorithm 2: Sampling from a typical GMRF-based full conditional encountered in block Gibbs sampling (Rue and Held, 2005).

In single-site Gibbs sampling, we sequentially draw from each univariate distribution with density $p(x_i \mid x_{(-i)})$, $i = 1, \dots, n$. This requires using $x_{(-i)}$ to calculate η_i prior to drawing from each of the n conditional distributions, meaning that single-site updating essentially becomes an $\mathcal{O}(n^2)$ operation. This algorithm has little regard for the ordering of the nodes, making such sampling strategies very easy to implement. Compared to block updating, though, many more Gibbs scans are required to sufficiently explore the support of the distribution. This approach is the most iteration-intensive of any of the approaches considered here. As such, its implementation in R can result in a large amount of computational overhead associated with loops, considerably slowing the entire routine.

Efficient block sampling schemes for GMRFs are discussed in Rue (2001) and Knorr-Held and Rue (2002). What most of these schemes have in common is the use of a Cholesky factorization of $Q = LL^T$ and using the factorization to solve a system of equations. For instance, an algorithm presented by Rue and Held (2005) for block sampling from the unconditional (prior) GMRF in (2) is given in Algorithm 1. The algorithm requires one Cholesky factorization and one linear solve via forward or backward substitution. For the case typically encountered in a Gibbs sampler, Rue and Held (2005) present the approach given in Algorithm 2 for simulating from conditional distributions of this form. This algorithm requires one Cholesky factorization and three linear solves via forward or backward substitution.

In general, for a matrix of dimension $n \times n$, the Cholesky factorization is an $\mathcal{O}(n^3/3)$ operation and each linear solve costs $\mathcal{O}(n^2)$ flops (Golub and Van Loan, 1996). This can be particularly onerous in a fully Bayesian approach in which hyperpriors are assigned to hyperparameters θ that appear in the precision matrix $Q_p \equiv Q_p(\theta)$. In such models we cannot simply compute the Cholesky factorization once and store it for repeated use. The factorizations and linear solves must be carried out on each full sweep of the sampler.

The key to making block updating feasible on high-dimensional data lies in the computational savings that can be achieved when \mathbf{Q} is sparse. By sorting the rows and columns of \mathbf{Q} to have a banded structure, the Cholesky factorization reduces to an approximately $\mathcal{O}(np^2)$ operation for a matrix with bandwidth $p \ll n$. The sparsity of \mathbf{Q} will be inherited by the Cholesky factor \mathbf{L} , so that the factor has lower bandwidth p (Rue and Held, 2005, Theorem 2.9). The consequence is that the cost of solving the system $\mathbf{L}^T \mathbf{v} = \mathbf{z}$ in step 5 of Algorithm 2 is reduced to $\mathcal{O}(np)$ when $n \gg p$. However, finding a low-bandwidth structure for \mathbf{Q} requires finding an optimal permutation of the node labels, which itself is a non-trivial problem requiring specialized knowledge beyond the expertise of many statisticians. Indeed, concerning this point, Rue and Held (2005, p. 52) recommend “leaving the issue of constructing and implementing algorithms for factorizing sparse matrices to the numerical and computer science experts.” In practice, most statisticians will rely on special functions for sparse matrices such as those found in the **Matrix** package in **R** (Bates and Maechler, 2016).

Sparsity of the precision matrix is the typical case when sampling from a distribution of the form (2). For simulating posterior distributions via block Gibbs sampling, however, we are interested in drawing from full conditional distributions as in (3). In this case, sparsity of the entire precision matrix is contingent upon the sparsity of $\mathbf{A}^T \boldsymbol{\Sigma}^{-1} \mathbf{A}$. Fortunately, this is often the case in practice. For instance, in disease mapping and related applications, it is common to place a spatially correlated random effect at each location to encourage smoothing of the incidence rate over space (e.g., Waller et al., 1997; Banerjee et al., 2015). In terms of the linear model, this can be expressed as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon} \Leftrightarrow \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \equiv \mathbf{y}^* = \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}$, where $\mathbf{X}\boldsymbol{\beta}$ corresponds to fixed effects and $\boldsymbol{\gamma}$ contains the spatially-varying effects. With site-specific random effects, \mathbf{Z} is diagonal or block diagonal. The diagonal case (e.g., $\mathbf{Z} = \mathbf{I}$) is especially amenable to efficient block Gibbs sampling and the chromatic sampler discussed in Subsection 2.2, since the underlying graph \mathcal{G} for the full conditional distribution is exactly the same as the prior graph. For general Bayesian linear models in which $\mathbf{A}^T \boldsymbol{\Sigma}^{-1} \mathbf{A}$ is dense, single-site sampling might be preferable to block Gibbs sampling unless the matrix has a rapidly decaying spectrum which can be well approximated with low-rank matrices (Brown et al., 2016; Fox and Norton, 2016).

Even with sparse matrix algebra, block sampling \mathbf{x} from a GMRF in very high dimensions can still be problematic (Rue, 2001, p. 331). As a consequence, Rue (2001) proposed a blocking scheme for updating the random field over one subset of nodes at a time. If the $n_1 \times n_1$ lattice is partitioned into blocks each of dimension $n_1/c \times n_1/c$, for some c , then the effort of Cholesky factorization is reduced by a factor of $1/c^2$. This again leads to the problem of finding a good permutation of the nodes and may result in serial computations to update the entire field. The chromatic Gibbs sampler discussed in Subsection 2.2 is closely related to this partitioning approach. The idea in chromatic Gibbs is to partition the nodes according to a graph coloring whereby each subset is conditionally independent and can be updated simultaneously.

2.2 Chromatic Gibbs Sampling

Again considering the graph representation of the GMRF, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the *local Markov property* says that $x_i \perp \mathbf{x}_{-(i, \mathcal{N}(i))} \mid \mathbf{x}_{\mathcal{N}(i)}$, where $\mathbf{x}_{-(i, \mathcal{N}(i))}$ denotes all x except x_i and the neighborhood of x_i , and \perp denotes (statistical) independence. An extension of the local Markov property is to let $C \subset \mathcal{V}$ denote a separating set, or *cut*, of \mathcal{G} such that nodes in a set $A \subset \mathcal{V}$ are disconnected from nodes in $B \subset \mathcal{V}$ after removing the nodes in C from the graph. Then the *global Markov property* states that $\mathbf{x}_A \perp \mathbf{x}_B \mid \mathbf{x}_C$.

A *coloring* $f : \mathcal{V} \rightarrow \{1, \dots, k\}$, $k \in \mathbb{N}$, is a collection of labels assigned to nodes on a graph so that no two nodes that share an edge have the same label. A k -coloring induces a partition of

the nodes $\{\mathcal{A}_1, \dots, \mathcal{A}_k\}$, where $\mathcal{A}_j = f^{-1}(\{j\}) \subset \mathcal{V}$. For example, Figure 1 displays a 4-coloring that could be used for data that lie on a regular two-dimensional lattice. Such data are commonly encountered in imaging analysis. Given a k -coloring of the MRF graph, we can determine a cut C_j corresponding to each color j by assigning all nodes that are not of that color to be in the cut; i.e., $C_j = \mathcal{A}_j^c$, $j = 1, \dots, k$. Defining cuts in this way for $j = 1, \dots, k$, we have that

$$x_i \mid \mathbf{x}_{C_j} \stackrel{\text{indep.}}{\sim} N(\eta_i, \sigma_i^2), \quad i \in \mathcal{A}_j,$$

where each η_i and σ_i^2 depends on elements of \mathbf{x}_{C_j} . In other words, all nodes of the same color are conditionally independent and hence can be updated in parallel, given the rest of the field. The use of graph colorings to exploit the local Markov property and thus facilitate parallel sampling leads Gonzalez et al. (2011) to term the approach *chromatic Gibbs sampling*.

⊗	⊕	⊗	⊕	...	⊗	⊕
⊙	⊙	⊙	⊙	...	⊙	⊙
⊗	⊕	⊗	⊕	...	⊗	⊕
⊙	⊙	⊙	⊙	...	⊙	⊙
⋮	⋮	⋮	⋮	⋮	⋮	⋮
⊗	⊕	⊗	⊕	...	⊗	⊕
⊙	⊙	⊙	⊙	...	⊙	⊙

Figure 1: An example of a k -coloring ($k = 4$) for nodes on a regular two-dimensional lattice. Each symbol represents a different label.

Algorithm 3 presents the general chromatic Gibbs sampler. An advantage of using this approach is seen in step 3 of the algorithm. When the updating of the random variables indexed by \mathcal{A}_j is distributed across several processors, the computational effort of updating the entire field can be dramatically reduced, even compared to the approximate linear complexity obtained from sparse matrix factorization. Indeed, given p processors and a k -coloring of a Markov random field over n nodes, the chromatic Gibbs sampler generates a new sample in approximately $\mathcal{O}(n/p + k)$ operations (Gonzalez et al., 2011, p. 326). Algorithm 3 facilitates the simultaneous updating of multiple sites at once like a block Gibbs sampling algorithm while using only a set of univariate full conditional distributions as in single-site Gibbs sampling. In this sense, the chromatic Gibbs sampler can be regarded as a hybrid sampling approach. Of course, the best computational savings will be achieved by using the chromatic index for the coloring (i.e., the minimum k so that a k -coloring of \mathcal{G} exists), but the minimal coloring problem for a graph is NP-Complete and thus very challenging except in simple situations. Any coloring, though, with $k < n$ will result in an improvement. On regular lattices with commonly assumed neighborhood structures (e.g., Figure 1), such colorings can be found by inspection without complicated algorithms. In Section 3, we give a greedy algorithm for

finding a coloring on an arbitrary graph, including irregular lattices that would be encountered in, e.g., disease mapping and climate studies. Hence the chromatic Gibbs sampling approach should be accessible to practitioners who lack any deep knowledge of graph theory.

Input: Current state of GMRF, $\mathbf{x}^{(t)}$, a k -coloring of the MRF graph, $\{\mathcal{A}_j : j = 1, \dots, k\}$.
Output: New draw $\mathbf{x}^{(t+1)}$ from the GMRF.

```

1 for  $j = 1$  to  $k$  do
2   For  $i \in \mathcal{A}_j$ , calculate conditional means and standard deviations  $\eta_i, \sigma_i^2$  using
      $\mathbf{x}_{\mathcal{A}_1}^{(t+1)}, \dots, \mathbf{x}_{\mathcal{A}_{j-1}}^{(t+1)}, \mathbf{x}_{\mathcal{A}_{j+1}}^{(t)}, \dots, \mathbf{x}_{\mathcal{A}_k}^{(t)}$ 
3   Draw  $\mathbf{x}_{\mathcal{A}_j} \sim N \left[ (\eta_1, \dots, \eta_{|\mathcal{A}_j^c|})^T, \text{diag}(\sigma_i^2, i = 1, \dots, |\mathcal{A}_j^c|) \right]$ 
4 end
5 Return  $\mathbf{x}^{(t+1)}$ 

```

Algorithm 3: Chromatic Gibbs step for updating a GMRF inside an MCMC algorithm (Gonzalez et al., 2011).

For large-scale computing environments in which the number of available processors $p = \mathcal{O}(n)$, chromatic Gibbs updating takes $\mathcal{O}(k)$ operations, a dramatic improvement when $k \ll n$. For many practitioners, of course, such resources will be unavailable. Most computers today have parallel processing capabilities, though, and any distributed processing over p processors can reduce the computational burden by an approximate factor of $1/p$. It is important to note that while we update *within* each color simultaneously, we still update *between* colors serially. This way, we guarantee an appropriate Gibbs updating schedule for each node on the lattice and the algorithm converges to the correct target distribution.

Regardless of the number of processors available to the user, savings can still be realized when working in a high-level language such as **R** by ‘vectorizing’ the updating of the conditionally independent sets. Vectorizing still ultimately uses a **for** loop on each set of nodes, but the loops are performed in a faster language such as **C** or **Fortran**. It also minimizes the overhead associated with interpreting data types; i.e., vectorizing allows **R** to interpret the data type only once for the entire vector instead of repeatedly for each element of the vector, as is the case when looping in **R**.

Gonzalez et al. (2011) propose also a “splash sampler” to combine the blocking principle of updating sets of correlated variables together with the parallelizability afforded by graph colorings. However, it may not be obvious how to effectively partition the nodes so that groups of highly correlated variables are kept together while minimizing the dependence between groups. The splash sampler adaptively determines such groups by using sophisticated searching algorithms to construct undirected acyclic graphs called junction trees. The tree construction can itself be computationally intensive and delicate to effectively implement, and the computational complexity of the subsequent parallel sampler depends on the size of the largest tree. Our experience has been that despite the slower statistical efficiency of not updating correlated blocks simultaneously, the gain in computational efficiency from the simple chromatic sampler can still outweigh the loss of sampling efficiency, leading to an overall improvement in cost per effective (i.e., independent) sample from the Markov chain. This fact, combined with the much simpler implementation for practitioners, leads us to advocate for the chromatic sampler over the more sophisticated alternative.

In the sequel, we illustrate improvements over both block Gibbs sampling and single-site sampling that is afforded by chromatic Gibbs sampling, wherein we attain essentially identical results with much less computational effort. We emphasize that these improvements are gained *without* direct parallel processing. We simply vectorize the simultaneous updating steps, relieving **R** of the

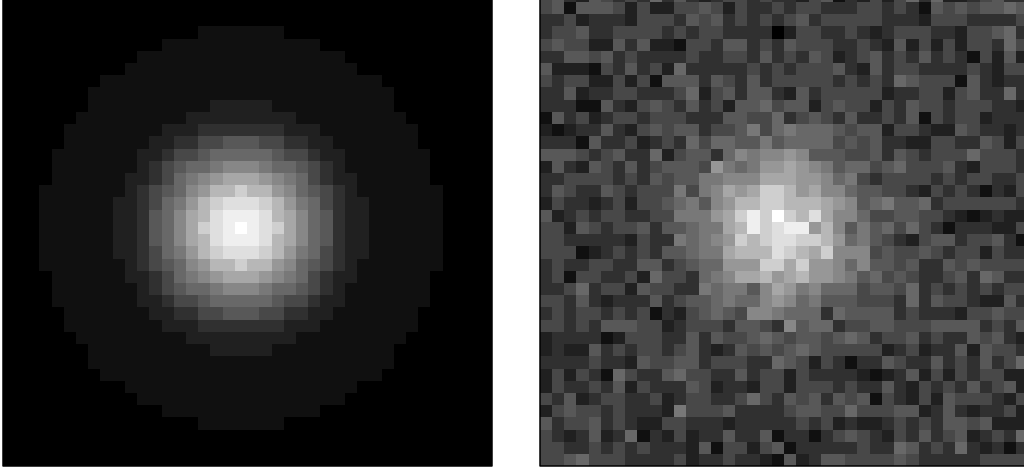


Figure 2: True image (left panel) and corrupted image (right panel) for the simulated image reconstruction example. (These particular images have resolution 40×40 .)

burden of direct `for` loops or matrix factorizations.

3 Numerical Illustrations

In this Section, we use two examples to illustrate the advantages of the chromatic sampler with respect to the two traditional sampling strategies, block and single-site Gibbs sampling. The first example is simulated data arranged according to a regular lattice as encountered in image analysis. The second example considers a Gaussian Markov random field over the irregular lattice formed by the counties of the United States of America.

3.1 Simulated Image Restoration

In general, the problem of image restoration involves attempting to reconstruct a true latent image, where the available data consist of pixel values according to color, often on the grayscale taking integer values from 0 to 255. The true values are assumed to have been contaminated with error. This area was one of the original motivating applications for Markov random fields (Besag, 1986). We consider an image consisting of $p \times p$ pixels, each of which has an observed value $y_{ij} = x_{ij} + \varepsilon_{ij}$, where x_{ij} is the true value of the $(i, j)^{\text{th}}$ pixel in the latent image and ε_{ij} represents the corresponding contamination. To generate data, we take the error terms to be independent, identically distributed $N(0, 0.1^2)$ random variables. The true image in this case is a rescaled bivariate Gaussian density with $x_{ij} = 5 \exp\{-\|\mathbf{v}_{ij}\|^2/2\}/\pi$, where $\mathbf{v}_{ij} = (v_i, v_j) \in [-3, 3] \times [-3, 3]$ denotes the center of the $(i, j)^{\text{th}}$ pixel, evenly spaced over the grid, and $\|\cdot\|$ denotes the usual Euclidean norm. Figure 2 depicts the true generated image (in 40×40 resolution) and its corrupted counterpart. To study the computational costs of each of the three sampling algorithms and how they scale with dimension $n = p^2$, we consider the same image collected at resolutions $p = 25, 50, 75, 100$.

The assumed model for the observed image is given by

$$\mathbf{y} = \mathbf{1}\beta_0 + \boldsymbol{\gamma} + \boldsymbol{\varepsilon}, \quad (4)$$

where $\mathbf{y} \in \mathbb{R}^n$ is the vector of the observed pixel values, $\mathbf{1}$ is the n -dimensional vector of ones,

$\beta_0 \in \mathbb{R}$ is a constant intercept parameter, $\boldsymbol{\gamma}$ is the vector of spatial effects, and $\boldsymbol{\varepsilon}$ is the vector of errors assumed to follow a $N(\mathbf{0}, \sigma^2 \mathbf{I})$ distribution. To capture local homogeneity of the images, we assume the spatial random effects obey an intrinsic autoregressive (IAR) model; i.e., $\boldsymbol{\gamma} \sim N(\mathbf{0}, \tau^2(\mathbf{D} - \mathbf{W})^{-1})$, where $\mathbf{W} = \{w_{ij} := I(i \sim j)\}_{i,j=1}^n$ is the incidence matrix of the underlying graph and $\mathbf{D} = \text{diag}(\sum_{j=1}^n w_{ij} : i = 1, \dots, n)$. Here we assume each pixel has a first-order neighborhood structure in which each interior pixel has eight neighbors. We ignore edge effects induced by the perimeter pixels of the image.

To complete the model formulation, we specify inverse gamma priors for the variance components and a flat prior was for the intercept; $\sigma^2 \sim IG(\alpha, \alpha)$, $\tau^2 \sim IG(\alpha, \alpha)$, and $\pi(\beta_0) \propto 1$. To approximate vague priors for the variance components, we take $\alpha = 0.001$. It has been observed that an inverse gamma prior on τ^2 sometimes can yield undesirable behavior in the posterior (Gelman, 2006), but our focus is on sampling the random field and thus we use this prior simply for convenience. For posterior sampling, our modeling assumptions lead to a Gibbs sampler having the following full conditional distributions:

$$\begin{aligned}\beta_0 | \mathbf{y}, \boldsymbol{\gamma}, \sigma^2 &\sim N(\mathbf{1}^T(\mathbf{y} - \boldsymbol{\gamma})/n, \sigma^2/n) \\ \sigma^2 | \mathbf{y}, \boldsymbol{\gamma}, \beta_0 &\sim IG(\alpha + n/2, \alpha + \|\mathbf{y} - \mathbf{1}\beta_0 - \boldsymbol{\gamma}\|^2/2) \\ \tau^2 | \boldsymbol{\gamma} &\sim IG(\alpha + n/2, \alpha + \boldsymbol{\gamma}^T(\mathbf{D} - \mathbf{W})\boldsymbol{\gamma}/2) \\ \boldsymbol{\gamma} | \mathbf{y}, \sigma^2, \tau^2 &\sim N(\mathbf{Q}^{-1}\mathbf{b}, \mathbf{Q}^{-1}),\end{aligned}$$

where $\mathbf{Q} = \sigma^{-2}\mathbf{I} + \tau^{-2}(\mathbf{D} - \mathbf{W})$ and $\mathbf{b} = (\mathbf{y} - \mathbf{1}\beta_0)/\sigma^2$.

For ease of notation, we revert to single-index notation for elements of \mathbf{y} , etc. as in Section 2. To implement the Gibbs sampler, three separate sampling strategies are employed, with the only difference being the treatment of $\boldsymbol{\gamma}$. First, since \mathbf{Q} is sparse, we consider full block Gibbs (FBG) sampling based on Algorithm 2 in Section 2 to sample $\boldsymbol{\gamma}$ in a single block. This approach makes use of the sparse matrix algebra routines available in the **Matrix** package in R (Bates and Maechler, 2016). The second strategy is single-site Gibbs (SSG) sampling in which we exploit the convenient full conditional distributions, $\gamma_i | \boldsymbol{\gamma}_{(-i)}, \mathbf{y}, \sigma^2, \tau^2 \sim N(\mu_i, \sigma_i^2)$, $i = 1, \dots, n$, where

$$\begin{aligned}\mu_i &= \frac{\tau^2 y_i + \sigma^2 \sum_{j \in \mathcal{N}(i)} w_{ij} \gamma_j}{\sigma^2(\mathbf{D})_{ii} + \tau^2} \\ \sigma_i^2 &= \frac{\tau^2 \sigma^2}{\sigma^2(\mathbf{D})_{ii} + \tau^2}.\end{aligned}\tag{5}$$

In the single-site sampler, the sampling of $\boldsymbol{\gamma}$ is done element-wise, obviating the need to work with large matrices. The final sampling strategy we implement is chromatic Gibbs (CBG) sampling discussed in Subsection 2.2. This approach uses the coloring depicted in Figure 1 as a 4-coloring of the pixels in the image. Following the notation in Subsection 2.2, we have that

$$\gamma_i | \boldsymbol{\gamma}_{C_j}, \mathbf{y}, \sigma^2, \tau^2 \stackrel{\text{indep.}}{\sim} N(\mu_i, \sigma_i^2), \quad i \in \mathcal{A}_j, \quad j = 1, \dots, 4,$$

where μ_i and σ_i^2 are defined in (5). It is worthwhile to note that all of the necessary conditional means and variances for a given color can be computed simultaneously through matrix multiplication and addition, as opposed to the iterative updating of these quantities in the single-site sampler. Moreover, this approach does not make use of time consuming computations associated with the Cholesky factorization of large matrices or solving high dimensional systems, which is necessary for FBG. The most important feature here, however, is that the γ_i can be drawn simultaneously. A full update of $\boldsymbol{\gamma}$ can essentially be completed in 4 steps instead of the p^2 steps required for SSG.

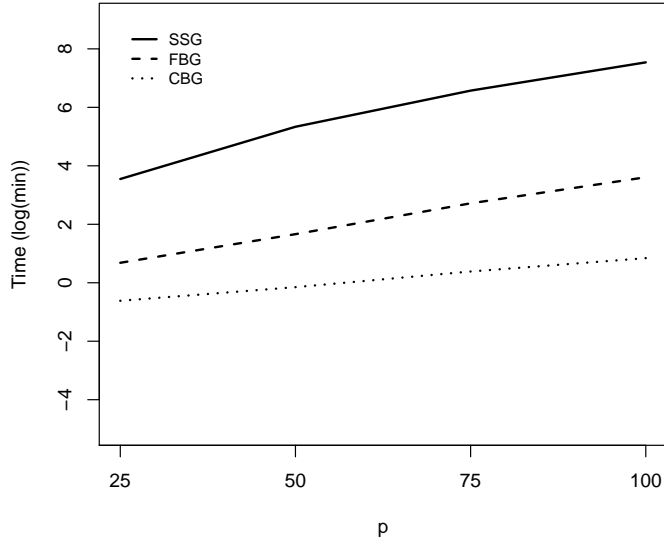


Figure 3: Computational time (in $\log(\text{min})$) required to complete 20,000 Gibbs iterates versus image size ($p \times p$) for the simulated image restoration example using chromatic Gibbs sampling (CBG), full block Gibbs (FBG), and single-site sampling (SSG).

We implement the three sampling strategies so that each procedure draws $B = 20,000$ realizations from the Monte Carlo Markov chain to approximate the posterior distribution of the model parameters. We repeated the sampling over successively refined target image resolutions, with 25×25 , 50×50 , 75×75 , and 100×100 . Hence the dimensions of γ and the underlying MRF are $n = 625$, 2500 , 5625 , 10000 . The simulations, coded entirely in R, are carried out on a Dell Optiplex 790 desktop running Windows 7 with an Intel Core i7-2600 3.40 GHz CPU and 16 GB of RAM.

Figure 3 displays the total wall time (in $\log(\text{min})$) required by each algorithm to complete 20,000 iterations. There are notable differences between the algorithms even at the coarsest resolution. However, the most pronounced differences are seen at the finest resolution. At this level, the time differences are on orders of magnitude. We can see that the block sampler does well with approximately 37 minutes needed, whereas the single-site sampler takes well over 31 hours. Both of these approaches are much slower than the chromatic Gibbs sampler, which needed only two and half minutes to complete the same number of iterations.

It is well-known that different MCMC algorithms have different rates of convergence, so that even using the same number of samples from two algorithms is not guaranteed to provide the same quality of posterior approximation. To accommodate the different convergence characteristics of the three algorithms while still considering total computation time, we measure also the *cost per effective sample* (Fox and Norton, 2016),

$$CES := \frac{\kappa\tau}{N},$$

where τ is the total computation time, N is the length of the Markov chain, and κ is the integrated autocorrelation time (Kass et al., 1998; Carlin and Louis, 2009). This quantity measures the total computational effort required to generate an effectively *independent* sample from the target distribution. Figure 4 shows the measured *CES* values (in $\log(\text{s})$) for each sampling algorithm over all of

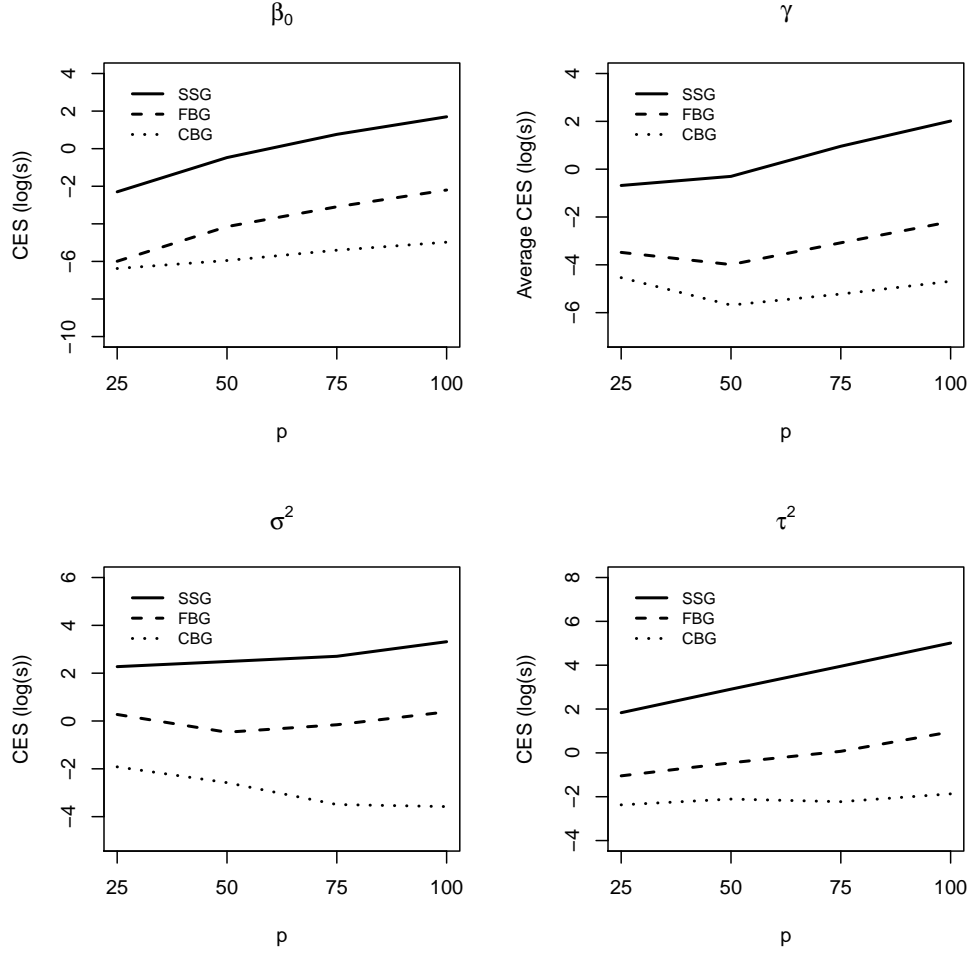


Figure 4: Costs per effective sample (in log(sec)) required for chromatic Gibbs sampling (CBG), full block Gibbs (FBG), and single-site sampling (SSG) versus resolution ($p \times p$ image). The values plotted for γ are the average CES over all γ_i .

the resolutions. We again observe substantial differences between the sampling approaches that become more pronounced as resolution increases. In each instance, chromatic sampling again requires only a fraction of the computational effort to generate an independent sample when compared to FBG or SSG. The most extreme disparities are seen in the chain for τ^2 at $p = 100$. Here, single-site sampling needs over two minutes to generate an independent sample. Block Gibbs is much better, requiring only two seconds per effective sample. Both of these procedures are by far outperformed by chromatic Gibbs sampling, however. The CBG approach only needs about one tenth of a second to generate a new independent sample from the approximate posterior distribution.

3.2 Temperature Mapping on an Irregular Lattice

Here we examine the performance of the three strategies on an irregular lattice, since both the structure of \mathbf{Q} and the possible colorings of the underlying graph can complicate sampling approaches. We use average county-level temperatures observed during the 2015 calendar year at each of the $n = 3109$ counties in the continental United States to reconstruct a smooth map of the average

<p>Input: MRF graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.</p> <p>Output: k-coloring partition $\{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_k\}$, for some k.</p> <pre> 1 Set $j = 1$ and $\mathcal{A}_0 = \emptyset$ 2 while $\mathcal{V} \setminus \bigcup_{l=0}^{j-1} \mathcal{A}_l \neq \emptyset$ do 3 $\mathcal{I}_j \leftarrow \mathcal{V} \setminus \bigcup_{l=0}^{j-1} \mathcal{A}_l$ 4 $\mathcal{A}_j \leftarrow \emptyset$ 5 while $\mathcal{I}_j > 0$ do 6 $i \leftarrow \min \mathcal{I}_j$ 7 $\mathcal{A}_j \leftarrow \mathcal{A}_j \cup \{i\}$ 8 $\mathcal{I}_j \leftarrow \mathcal{I}_j \setminus (\{i\} \cup \mathcal{N}(i))$ 9 end 10 $j \leftarrow j + 1$ 11 end 12 $k \leftarrow j - 1$ 13 Return $\{\mathcal{A}_1, \dots, \mathcal{A}_k\}$ </pre>
--

Algorithm 4: Greedy algorithm for k -coloring the nodes of an irregular lattice.

variation in annual temperatures over the entire United States.

We again consider model (4) as in Subsection 3.1. Here, though, y_i is the average yearly temperature observed in the i^{th} county, β_0 is the overall average temperature, γ_i is the spatially-correlated deviation from the average temperature for the i^{th} county, and ε_i is the error. Since we are again assuming that the error terms are Gaussian and the spatial random effects follow an IAR model, the sampling strategies outlined in the previous section can be directly implemented.

For the continental United States, we take counties that share a border to be neighbors. In order to implement CBG, a coloring of the graph corresponding to this neighborhood structure has to be found. On such an irregular lattice, finding a coloring simply by inspection is not easy. Moreover, finding the minimal coloring of the graph is not tractable. In spite of this, we still can find a possibly suboptimal coloring of the graph via the greedy algorithm given in Algorithm 4. Applying the algorithm, we are able to obtain a 7-coloring so that the chromatic sampler can update the entire $n = 3109$ -dimensional field in seven steps, each of which can update a subset of the field simultaneously. Figure 5 displays the coloring discovered via Algorithm 4.

We again implement each of the samplers considered in Section 3.1 to run 10,000 iterations of the Gibbs sampling Markov chain for all of the model parameters. Table 1 summarizes the results for all three samplers in terms of wall time and cost per effective sample of the intercept term. The estimated temperatures obtained under each sampling strategy are displayed in Figure 6. Evidently, we are able to get essentially identical results under each sampling strategy, with CBG sampler being the fastest to return them. These results reinforce the findings from our simulation described in Subsection 3.1. As before, the CBG sampler is far more computationally efficient when compared to FBG and SSG, even if the data are arranged according to an irregular lattice.

Our results suggest that the chromatic Gibbs sampling strategy is preferable when compared to the two most common strategies of block sampling via sparse linear algebra and single-site sampling via local characteristics. In simulated image reconstruction, we found that for every considered resolution, the chromatic sampler is computationally much cheaper than the full block Gibbs and single-site samplers. The cost per effective sample is also consistently smaller, suggesting sufficient convergence to the target distribution with much less computational effort. Moreover, as the dimension of the problem grows, the gains in computational efficiency become more pronounced. The

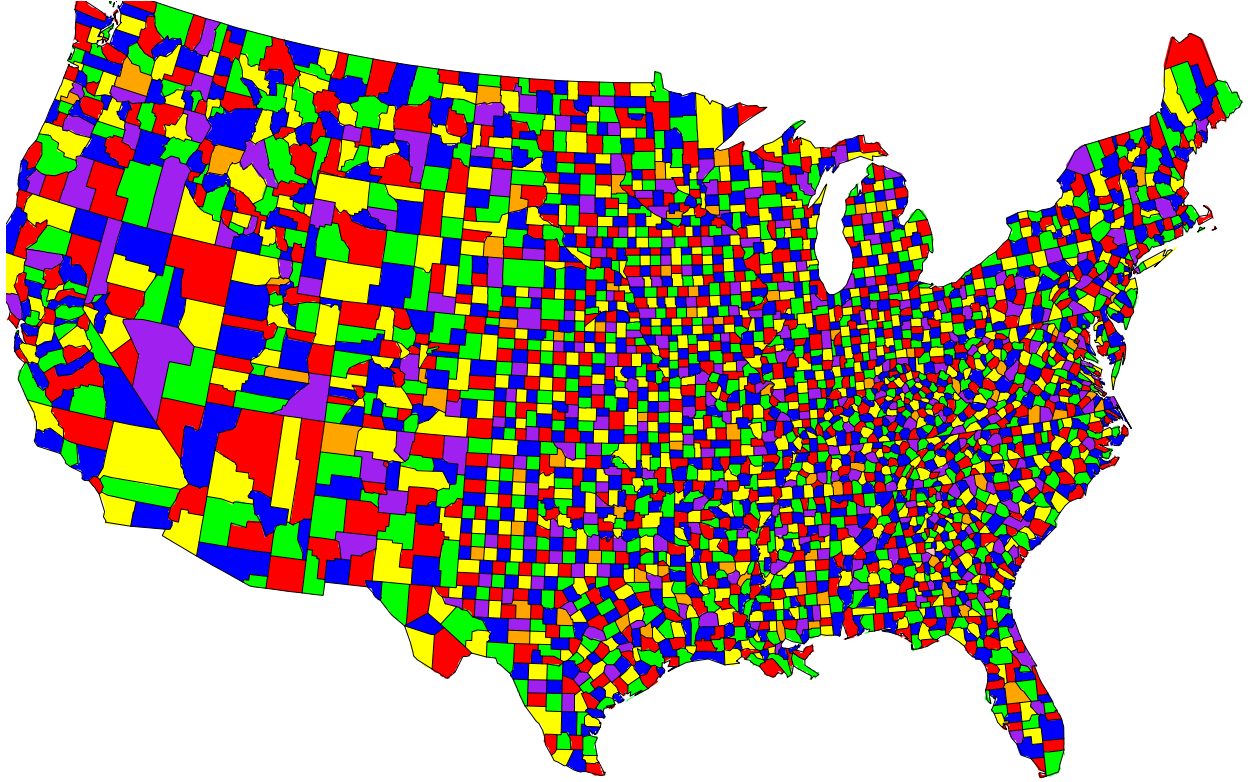


Figure 5: 7-Coloring of the $n = 3109$ counties in the continental United States found via Algorithm 4.

chromatic Gibbs sampler completes the sampling process roughly 4, 6, 10, and 15 times faster than even the full block Gibbs sampler for $p = 25, 50, 75, 100$, respectively. The single-site sampler is the worst performer, often taking more than 100 hundred times longer than chromatic Gibbs to complete the same number of iterations. We achieve similar gains when the data are on an irregular lattice, even with a coloring that is not guaranteed to be optimal. Once we can find a coloring with, say, a greedy algorithm, we are in a position to dramatically accelerate MCMC computations involving GMRFs.

	Algorithm		
	SSG	FBG	CBG
Wall Time	43.800	658.20	8982.6
β_0 CES	0.0009	0.0664	0.7270

Table 1: Wall time and cost per effective sample of the intercept β_0 (in seconds) for single-site Gibbs (SSG), full block Gibbs (FBG), and chromatic Gibbs (CBG) sampling in the United States temperature mapping example.

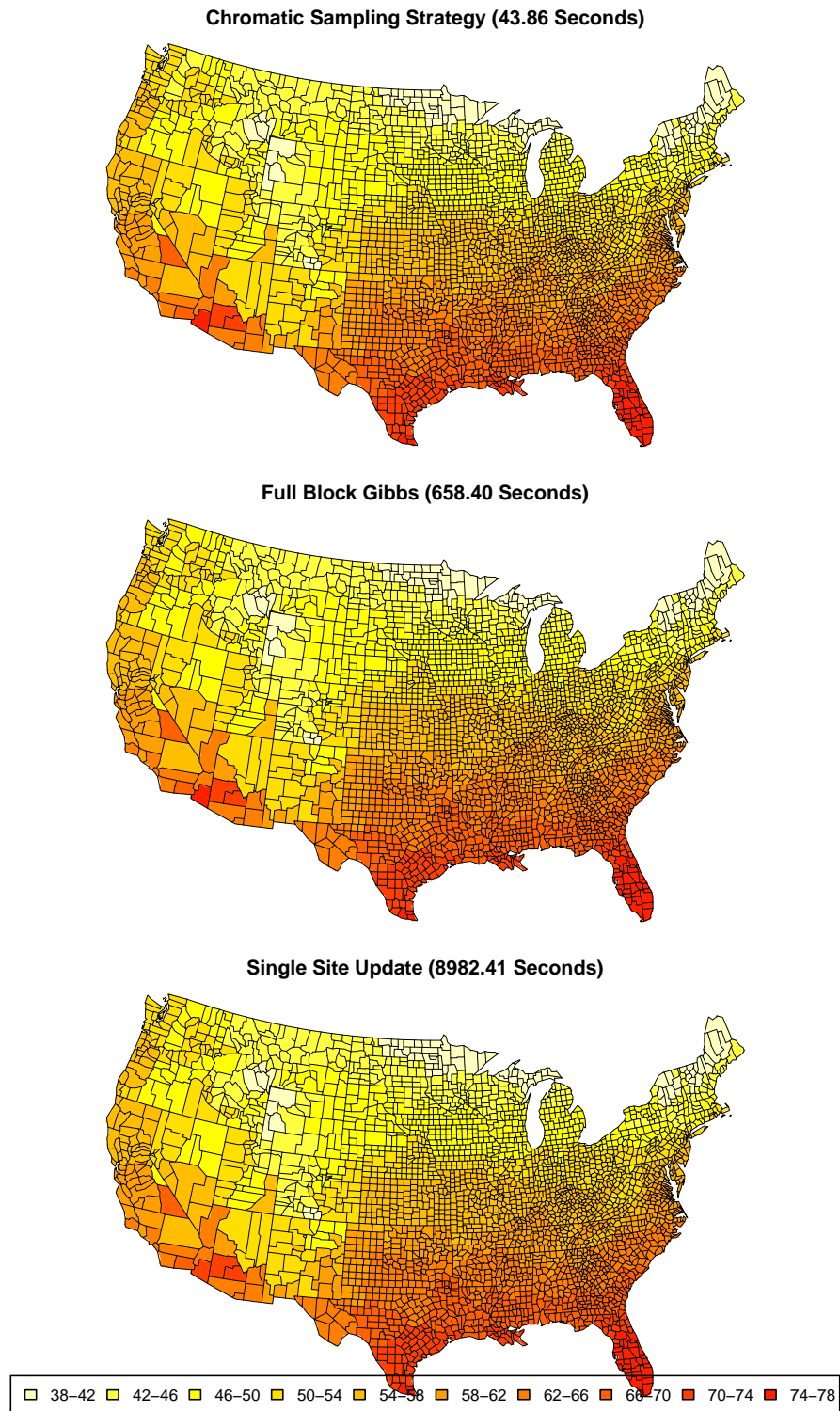


Figure 6: Estimated average annual temperatures over the continental United States obtained from the single-site (SSG), full block Gibbs (FBG), and chromatic Gibbs (CBG) sampling strategies with 10,000 iterations, along with wall time required to obtain the results.

4 Discussion

Over the last twenty years, Markov random fields and, in particular, Gaussian Markov random fields have seen a dramatic increase in popularity in the applied Bayesian community. In this work, we have discussed several approaches for efficiently simulating from Gaussian Markov random fields that are commonly used in practice. Our focus has been on the two dominant approaches in the statistics literature, namely single-site updating via the scalar-valued local characteristics and block updating the entire field at once. We compared these to chromatic Gibbs sampling, a more recent novelty to emerge from the machine learning literature. Each procedure has theoretical guarantees and convergence properties, but our criteria has been pragmatic; i.e., how can statistics practitioners effectively lower the computational cost of sampling from the target distribution without resorting to esoteric knowledge from graph theory, numerical analysis, or parallel programming? Taking this view, we find that chromatic Gibbs sampling is a suitable answer. We demonstrate numerically that chromatic sampling is competitive with and often more efficient than single-site and full block Gibbs. This performance holds on both regular and irregular lattices, the latter of which was demonstrated without using a provably optimal coloring of the MRF graph. Both block sampling and chromatic sampling tend to be far superior to single-site sampling when one is trying to avoid direct iterations in \mathbf{R} . Further, the relative computational savings of chromatic sampling versus block sampling can be dramatic as the dimension of the problem grows, even in the presence of sparse precision matrices. For widely-used Bayesian models with spatially-correlated but site-specific random effects, the concomitant Gibbs sampling algorithm lends itself nicely to chromatic sampling, since a coloring of the graph corresponding to the joint conditional distribution can be identified easily.

In addition to the classical linear model considered here, we expect that sampling for Bayesian generalized linear mixed models (GLMMs) with spatial or temporal random effects can also be accelerated using a similar chromatic strategy. For a GMRF γ representing random effects in a GLMM, an appropriate coloring of the underlying graph induces conditional independence in the prior, $\pi(\gamma_{\mathcal{A}_j} \mid \gamma_{\mathcal{C}_j}) = \prod_{i \in \mathcal{A}_j} \pi(\gamma_i \mid \gamma_{\mathcal{C}_j})$. But this conditional independence carries through to the conditional posterior since the likelihood contribution of each γ_i is conditionally independent of the rest; i.e., $\pi(\gamma_{\mathcal{A}_j} \mid \gamma_{\mathcal{C}_j}, \mathbf{y}) \propto f(\mathbf{y} \mid \gamma) \pi(\gamma_{\mathcal{A}_j} \mid \gamma_{\mathcal{C}_j}) = \prod_{i \in \mathcal{A}_j} f(y_i \mid \gamma_i) \pi(\gamma_i \mid \gamma_{\mathcal{C}_j})$. Thus, $\gamma_{\mathcal{A}_j}$ can be updated simultaneously with, e.g., the weighted least squares Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) presented by Gamerman (1997). This is possibly an interesting avenue for future investigation that we do not pursue here.

While it is an evidently effective approach for accelerating MCMC sampling with GMRFs, chromatic Gibbs sampling should not be viewed as a panacea. Indeed, part of what makes it so easy to implement is the readily-identified coloring scheme in the underlying graph. When using a sparse MRF prior on certain parameters in a Bayesian model, there is no guarantee that the conditional posterior precision matrix will share that same property. Even if the conditional precision matrix is sparse, the underlying graph may be complicated enough that finding a reasonable graph coloring is too difficult to be useful, so one might be better off using the block updating procedures advocated by Rue (2001) and Knorr-Held and Rue (2002). In the worst case scenario, the conditional posterior precision matrix is dense with no underlying low-rank structure, in which case the chromatic sampler is prohibitively difficult to set up, and block updating via Cholesky factorization and linear solves is simply too expensive. In this situation, we would argue that single site sampling can be used as a last resort, if only because it is the only feasible option left.

Given the current trajectory of modern data analysis, the utility of GMRFs is not likely to diminish anytime soon. However, with their use comes the need for efficient yet accessible sampling strategies to facilitate Bayesian posterior inference along with appropriate measures of uncer-

tainty. This area remains an active area of research among statisticians, computer scientists, and applied mathematicians. Fortunately, the increasingly interdisciplinary environment within which researchers are operating today makes it more likely that each significant advancement will be widely disseminated and understood by researchers from a wide variety of backgrounds. This is no doubt a promising trend which will ultimately benefit the broader scientific community as a whole.

5 Acknowledgements

This material is based upon work partially supported by the National Science Foundation under Grant DMS-1127914 to the Statistical and Applied Mathematical Sciences Institute. DAB is partially supported by Grant CMMI-1563435 from the National Science Foundation. CSM is partially supported by Grant R01 AI121351 from the National Institutes of Health.

References

- Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2015), *Hierarchical Modeling and Analysis for Spatial Data*, Boca Raton: Chapman & Hall/CRC, 2nd ed.
- Bardsley, J. M. (2012), “MCMC-based image reconstruction with uncertainty quantification,” *SIAM Journal on Scientific Computing*, 34, 1316–1332.
- Bates, D. and Maechler, M. (2016), *Matrix: Sparse and dense matrix classes and methods*, R package version 1.2-6.
- Besag, J. (1974), “Spatial interaction and the statistical analysis of lattice systems,” *Journal of the Royal Statistical Society, Series B*, 36, 192–236.
- (1986), “On the statistical analysis of dirty pictures,” *Journal of the Royal Statistical Society, Series B*, 48, 259–302.
- Besag, J. and Kooperberg, C. (1995), “On conditional and intrinsic autoregressions,” *Biometrika*, 82, 733–46.
- Brown, D. A., Datta, G. S., and Lazar, N. A. (2017), “A Bayesian generalized CAR model for correlated signal detection,” *Statistica Sinica*, to appear.
- Brown, D. A., Lazar, N. A., Datta, G. S., Jang, W., and McDowell, J. E. (2014), “Incorporating spatial dependence into Bayesian multiple testing of statistical parametric maps in functional neuroimaging,” *NeuroImage*, 84, 97–112.
- Brown, D. A., Saibaba, A. K., and Vallélian, S. (2016), “Computationally efficient Markov chain Monte Carlo methods for hierarchical Bayesian inverse problems,” ArXiv 1609.07180.
- Cai, Z. (2014), “Very large scale Bayesian machine learning,” Unpublished doctoral dissertation, Rice University, Department of Computer Science.
- Cai, Z., Jermaine, C., Vagena, Z., Logothetis, D., and Perez, L. (2013), “The pairwise Gaussian random field for high-dimensional data imputation,” *IEEE 13th International Conference on Data Mining (ICDM)*, 61–70.

- Carlin, B. P. and Louis, T. A. (2009), *Bayesian Methods for Data Analysis*, Boca Raton: Chapman & Hall/CRC, 3rd ed.
- Cheng, D., Cheng, Y., Liu, Y., Peng, R., and Teng, S.-H. (2015), “Efficient sampling for Gaussian graphical models via spectral sparsification,” in *Journal of Machine Learning Research: Proceedings of the 28th International Conference on Learning Theory*, pp. 364–390.
- Dempster, A. P. (1972), “Covariance selection,” *Biometrics*, 28, 157–175.
- Fox, C. and Norton, R. A. (2016), “Fast sampling in a linear-Gaussian inverse problem,” *SIAM/ASA Journal of Uncertainty Quantification*, 4, 1191–1218.
- Gamerman, D. (1997), “Sampling from the posterior distribution in generalized linear mixed models,” *Statistics and Computing*, 7, 57–68.
- Gelfand, A. E. and Smith, A. F. M. (1990), “Sampling-based approaches to calculating marginal densities,” *Journal of the American Statistical Association*, 85, 398–409.
- Gelman, A. (2006), “Prior distributions for variance parameters in hierarchical models,” *Bayesian Analysis*, 1, 515–533.
- Geman, S. and Geman, D. (1984), “Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721–741.
- Golub, G. H. and Van Loan, C. F. (1996), *Matrix Computations*, Baltimore: The Johns Hopkins University Press, 3rd ed.
- Gonzalez, J. E., Low, Y., Gretton, A., and Guestrin, C. (2011), “Parallel Gibbs sampling: From colored fields to thin junction trees,” in *Journal of Machine Learning Research: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 324–332.
- Hastings, W. (1970), “Monte Carlo sampling methods using Markov chains and their application,” *Biometrika*, 57, 97–109.
- Higdon, D. M. (1998), “Auxiliary variable methods for Markov chain Monte Carlo with applications,” *Journal of the American Statistical Association*, 93, 585–595.
- Hodges, J. S., Carlin, B. P., and Fan, Q. (2003), “On the precision of the conditionally autoregressive prior in spatial models,” *Biometrics*, 59, 317–322.
- Johnson, M., Saunderson, J., and Willsky, A. (2013), “Analyzing Hogwild parallel Gaussian Gibbs sampling,” in *Advances in Neural Information Processing Systems 26*, eds. Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., Curran Associates, Inc., pp. 2715–2723.
- Kass, R. E., Carlin, B. P., Gelman, A., and Neal, R. (1998), “Markov chain Monte Carlo in practice: A roundtable discussion,” *The American Statistician*, 52, 93–100.
- Knorr-Held, L. and Rue, H. (2002), “On block updating in Markov random field models for disease mapping,” *Scandinavian Journal of Statistics*, 29, 597–614.

- Lindgren, F., Rue, H., and Lindström, J. (2011), “An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach,” *Journal of the Royal Statistical Society, Series B*, 73, 423–498.
- Liu, J. S., Wong, W. H., and Kong, A. (1994), “Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes,” *Biometrika*, 81, 27–40.
- Liu, Y., Kosut, O., and Willsky, A. S. (2015), “Sampling from Gaussian Markov random fields using stationary and non-stationary subgraph perturbations,” *IEEE Transactions on Signal Processing*, 63, 576–589.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), “Equation of state calculations by fast computing machines,” *Journal of Chemical Physics*, 21, 1087–1091.
- Niu, F., Recht, B., Ré, C., and Wright, S. J. (2011), “Hogwild! A lock-free approach to parallelizing stochastic gradient descent,” in *Advances in Neural Information Processing Systems 24*, eds. Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., Curran Associates, Inc., pp. 693–701.
- R Core Team (2016), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
- Rue, H. (2001), “Fast sampling of Gaussian Markov random fields,” *Journal of the Royal Statistical Society, Series B*, 63, 325–338.
- Rue, H. and Held, L. (2005), *Gaussian Markov Random Fields*, Boca Raton: Chapman & Hall/CRC.
- Rue, H. and Tjelmeland, H. (2002), “Fitting Gaussian Markov random fields to Gaussian fields,” *Scandinavian Journal of Statistics*, 29, 31–49.
- Schabenberger, O. and Gotway, C. A. (2005), *Statistical Methods for Spatial Data Analysis*, Boca Raton: Chapman & Hall/CRC.
- Song, H.-R., Fuentes, M., and Ghosh, S. (2008), “A comparative study of Gaussian geostatistical models and Gaussian Markov random fields,” *Journal of Multivariate Analysis*, 99, 1681–1697.
- Waller, L. A., Carlin, B. P., Xia, H., and Gelfand, A. E. (1997), “Hierarchical spatio-temporal mapping of disease rates,” *Journal of the American Statistical Association*, 92, 607–617.
- Xiao, G., Reilly, C., and Khodursky, A. B. (2009), “Improved detection of differentially expressed genes through incorporation of gene location,” *Biometrics*, 65, 805–814.